PADMAVATHY BHAMIDIPATI

FD IT FDN 110 B Au 22: Foundations Of Programming: Python

11/13/2022

Assignment 05

Lists and Dictionaries, Improved Scripts and GitHub

What we learnt in Module 5

In this module, we continued to learn more about lists, started on Dictionaries, learnt how to improve our Scripts and about GitHub as our code and other documents' repository.

Some useful concepts learnt in this module and applied in Assignment 5 are as follows:

- **Lists:** a couple of in-built functions such as unpacking using the * (overloaded) operator, and how to conveniently unpack a list's contents using **print()**. We also learnt how to save list data to a file and how to retrieve the same from a file into a list. We learnt **split()** to generate a list out of a string and **strip()** to clean unnecessary spaces and newline characters from a string.
- **Dictionaries:** are mapping types that are similar to sequence types but are stored and accessed as **key: value** pairs, with each pair separated by commas (,) and enclosed with braces {}. They are immutable. While Strings and Numbers or Tuples can be keys, the limitation is that they can only contain Strings and Numbers or Tuples themselves. We learnt how to work with dictionaries, and learnt some built-in methods from the dictionary class in Python.
- **Improving Scripts*:** with the increase in the complexity of the code we write, it has become clearer why adherence to commonly agreed upon coding practices becomes necessary. So, this portion of our learning module taught us four techniques on improved code must be written:
  - **Separation of Concerns:** this is a design principle that separates a program into distinct sections that each address different concerns handled by it
  - **Functions:** is a grouping of a set of statements that allow us to access them via a given name
  - **Script Templates:** this section taught us how to create a desired header for our script when using Spyder
  - **Structured Error Handling:** taught us how to handle exceptions or anticipated errors such as 'divided by 0'
- In this module too, we continued to use Spyder as our IDE for development.
- We used GitHub as a tool for our code / document repository, which will enable us to learn how to collaborate in larger teams.

(* Please note that we were asked not to implement anything learnt under Improving Scripts just yet.)

Assignment 05

For assignment 5, we were tasked with extending last week's program (Assignment) to use a list of dictionaries instead of a list of lists. Also, in addition to giving the user the ability to add to a CD inventory, reading from it, writing into it, displaying the existing list and exiting the program when done, deleting an entry has been added to the functionality. The list of dictionaries are added to a file or retrieved from it, as applicable.

I applied the various learning modules of this section of the course to write and successfully execute a Python script that offers a comprehensive menu that a user can choose from, to do the afore-mentioned tasks. Snips of code execution are listed under the **Code Execution** section below – in both **Spyder** and on the **Terminal**. The code is detailed in the **Appendix** section.

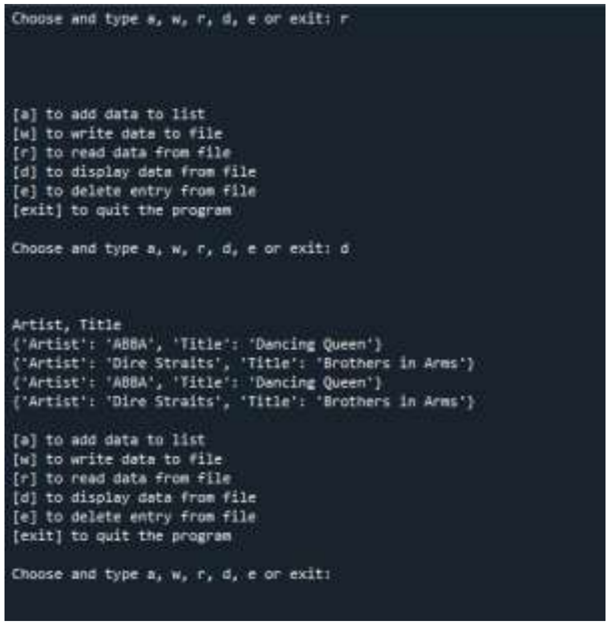
Assumptions, Challenges, Known Issues

Some constraints to writing more efficient code have been observed, probably since we are still in the early stages of learning Python.

- There is no check for duplicate entries anywhere in the program: duplicates may be introduced based duplicate entries being made by user themselves, and / or due to the order of user operations.

  For example, let two entries be made and written to a file. Close the program. Then, re-invoke the program and:
  CASE 1: display inventory >> no data displayed since in-memory list is empty
  CASE 2: read from the file first, then the data from the file is loaded into in-memory list. Any new entries are appended to the list and writing to the file now will render duplicates in the file.

  There are several such combinations as CASE 2 that could potentially introduce duplicates.



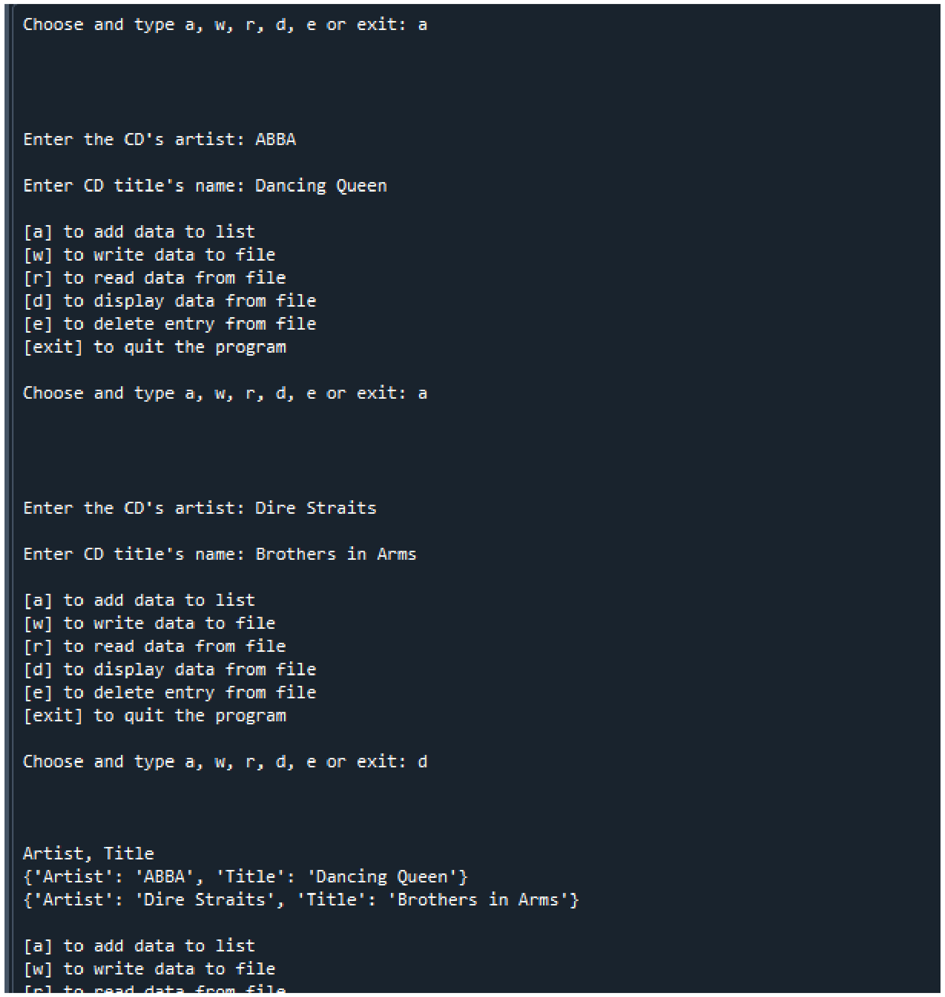
```
Choose and type a, w, r, d, e or exit: r



[a] to add data to list
[w] to write data to file
[r] to read data from file
[d] to display data from file
[e] to delete entry from file
[exit] to quit the program

Choose and type a, w, r, d, e or exit: d


Artist, Title
{'Artist': 'ABBA', 'Title': 'Dancing Queen'}
{'Artist': 'Dire Straits', 'Title': 'Brothers in Arms'}
{'Artist': 'ABBA', 'Title': 'Dancing Queen'}
{'Artist': 'Dire Straits', 'Title': 'Brothers in Arms'}

[a] to add data to list
[w] to write data to file
[r] to read data from file
[d] to display data from file
[e] to delete entry from file
[exit] to quit the program

Choose and type a, w, r, d, e or exit:
```

Figure 1: Example of duplicates being added due to in-memory list

- Also, there is no indication if the data in the in-memory list was saved to the file or not.

Code Execution in Spyder

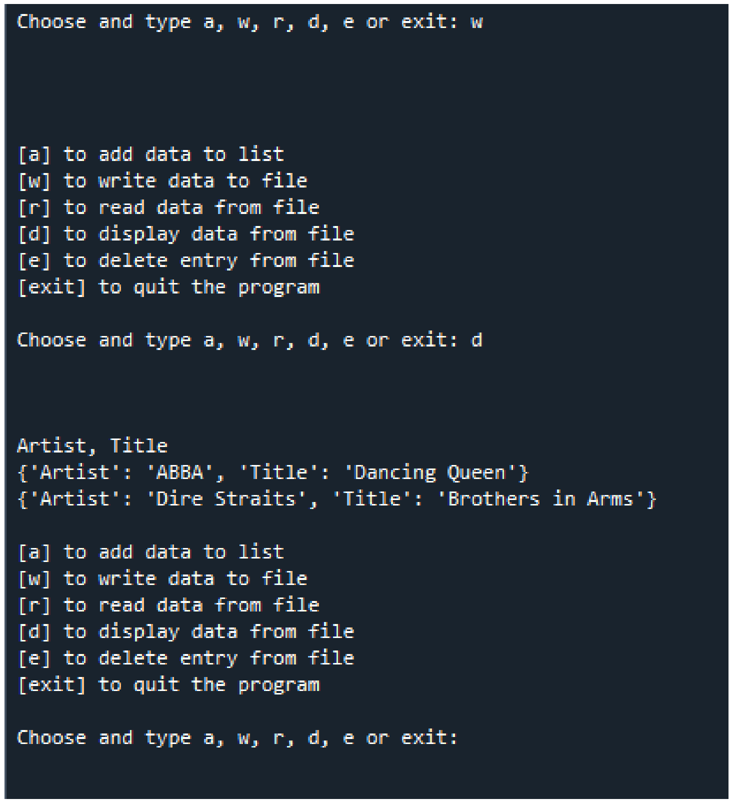
1. Adding entries to the CD Inventory



```
Choose and type a, w, r, d, e or exit: a




Enter the CD's artist: ABBA

Enter CD title's name: Dancing Queen

[a] to add data to list
[w] to write data to file
[r] to read data from file
[d] to display data from file
[e] to delete entry from file
[exit] to quit the program

Choose and type a, w, r, d, e or exit: a




Enter the CD's artist: Dire Straits

Enter CD title's name: Brothers in Arms

[a] to add data to list
[w] to write data to file
[r] to read data from file
[d] to display data from file
[e] to delete entry from file
[exit] to quit the program

Choose and type a, w, r, d, e or exit: d



Artist, Title
{'Artist': 'ABBA', 'Title': 'Dancing Queen'}
{'Artist': 'Dire Straits', 'Title': 'Brothers in Arms'}

[a] to add data to list
[w] to write data to file
```

Figure 2: Adding entries to CD Inventory and displaying from in-memory list

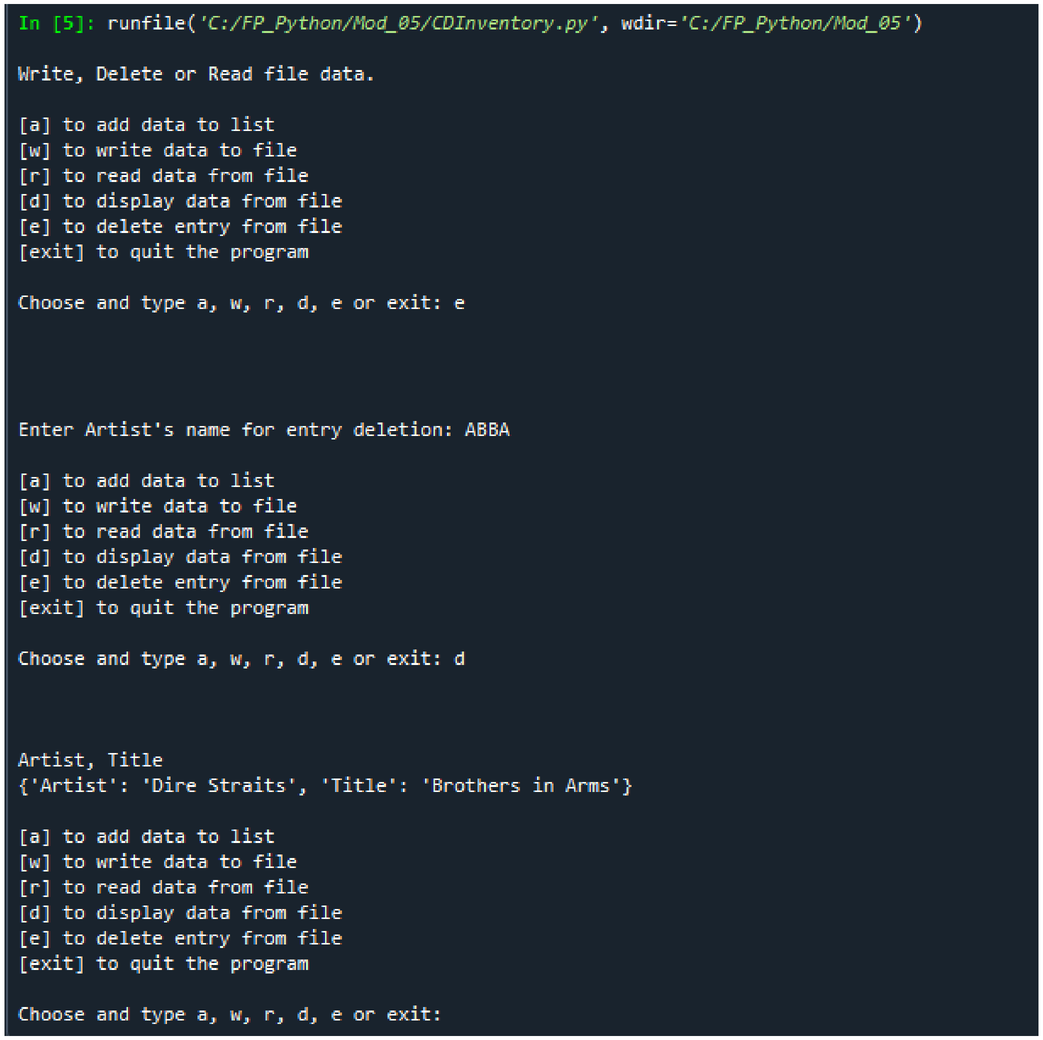
2. Writing entries to file



```
Choose and type a, w, r, d, e or exit: w




[a] to add data to list
[w] to write data to file
[r] to read data from file
[d] to display data from file
[e] to delete entry from file
[exit] to quit the program

Choose and type a, w, r, d, e or exit: d



Artist, Title
{'Artist': 'ABBA', 'Title': 'Dancing Queen'}
{'Artist': 'Dire Straits', 'Title': 'Brothers in Arms'}

[a] to add data to list
[w] to write data to file
[r] to read data from file
[d] to display data from file
[e] to delete entry from file
[exit] to quit the program

Choose and type a, w, r, d, e or exit:
```

Figure 3: Saving the entries to a file and displaying the same

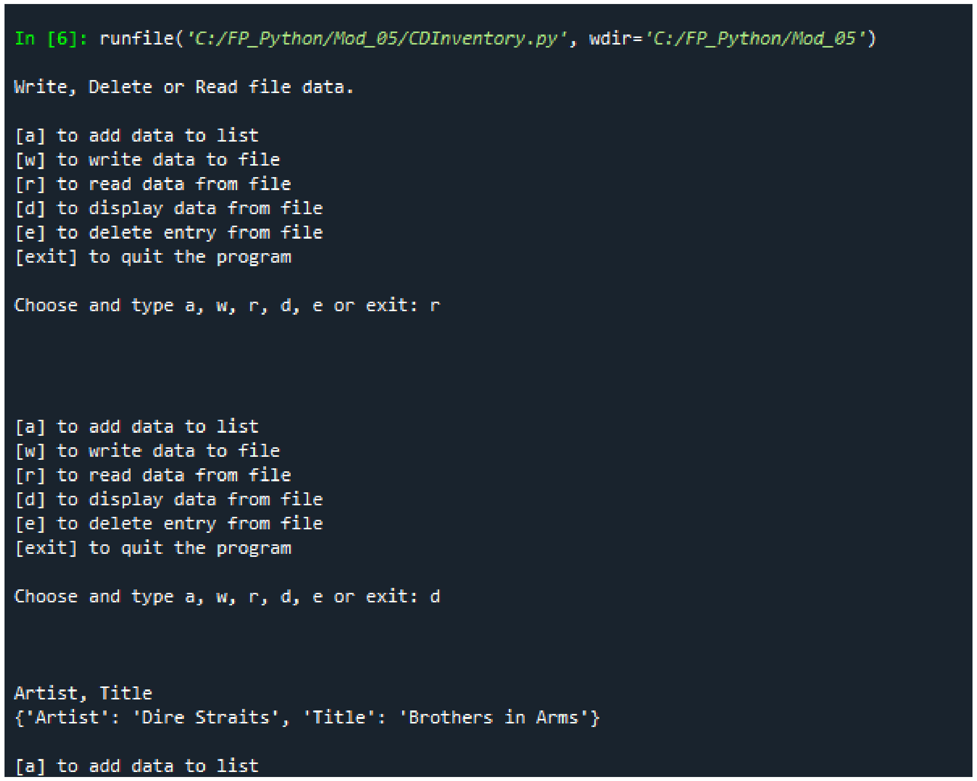
3. Deleting an entry made from file and displaying it



```
In [5]: runfile('C:/FP_Python/Mod_05/CDInventory.py', wdir='C:/FP_Python/Mod_05')

Write, Delete or Read file data.

[a] to add data to list
[w] to write data to file
[r] to read data from file
[d] to display data from file
[e] to delete entry from file
[exit] to quit the program

Choose and type a, w, r, d, e or exit: e




Enter Artist's name for entry deletion: ABBA

[a] to add data to list
[w] to write data to file
[r] to read data from file
[d] to display data from file
[e] to delete entry from file
[exit] to quit the program

Choose and type a, w, r, d, e or exit: d



Artist, Title
{'Artist': 'Dire Straits', 'Title': 'Brothers in Arms'}

[a] to add data to list
[w] to write data to file
[r] to read data from file
[d] to display data from file
[e] to delete entry from file
[exit] to quit the program

Choose and type a, w, r, d, e or exit:
```

Figure 4: Deleting an entry

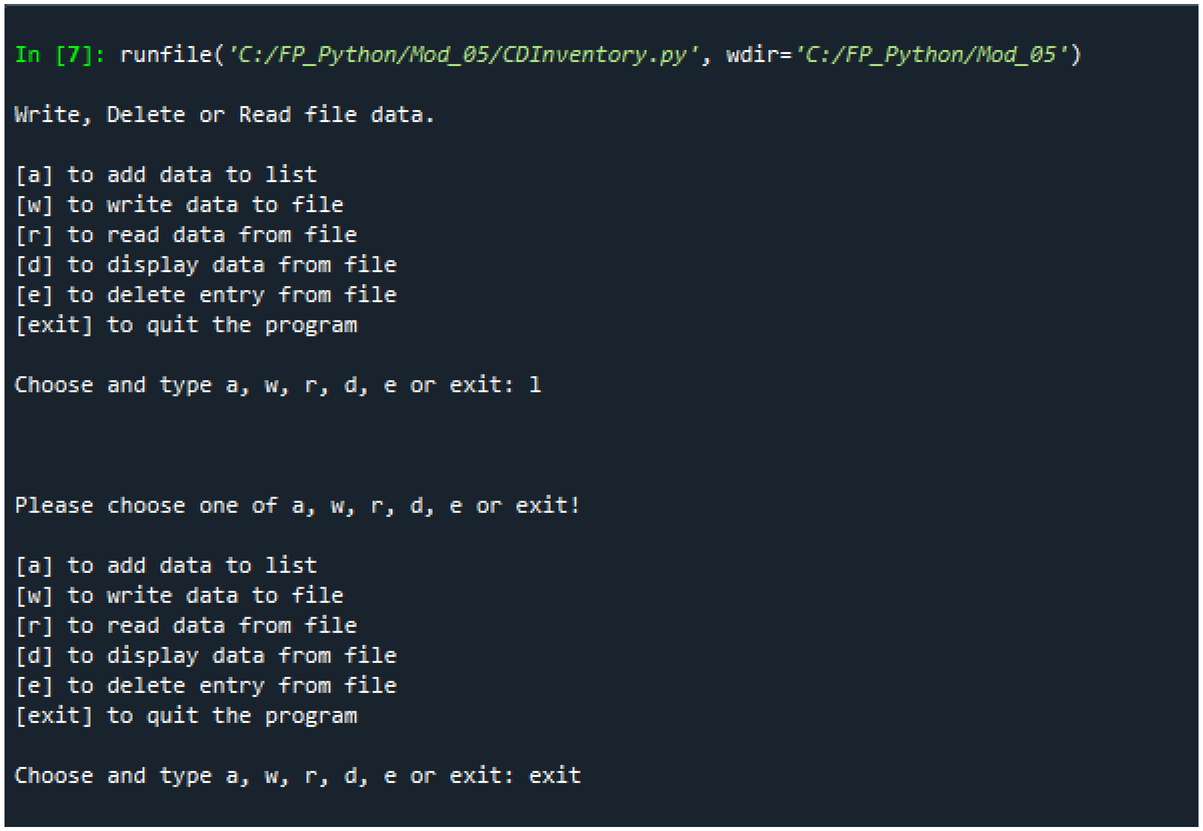
4. Reading data from file and displaying its contents



```
In [6]: runfile('C:/FP_Python/Mod_05/CDInventory.py', wdir='C:/FP_Python/Mod_05')

Write, Delete or Read file data.

[a] to add data to list
[w] to write data to file
[r] to read data from file
[d] to display data from file
[e] to delete entry from file
[exit] to quit the program

Choose and type a, w, r, d, e or exit: r




[a] to add data to list
[w] to write data to file
[r] to read data from file
[d] to display data from file
[e] to delete entry from file
[exit] to quit the program

Choose and type a, w, r, d, e or exit: d



Artist, Title
{'Artist': 'Dire Straits', 'Title': 'Brothers in Arms'}

[a] to add data to list
```

Figure 5: Reading current data from file and displaying it

5. Testing other menu options:

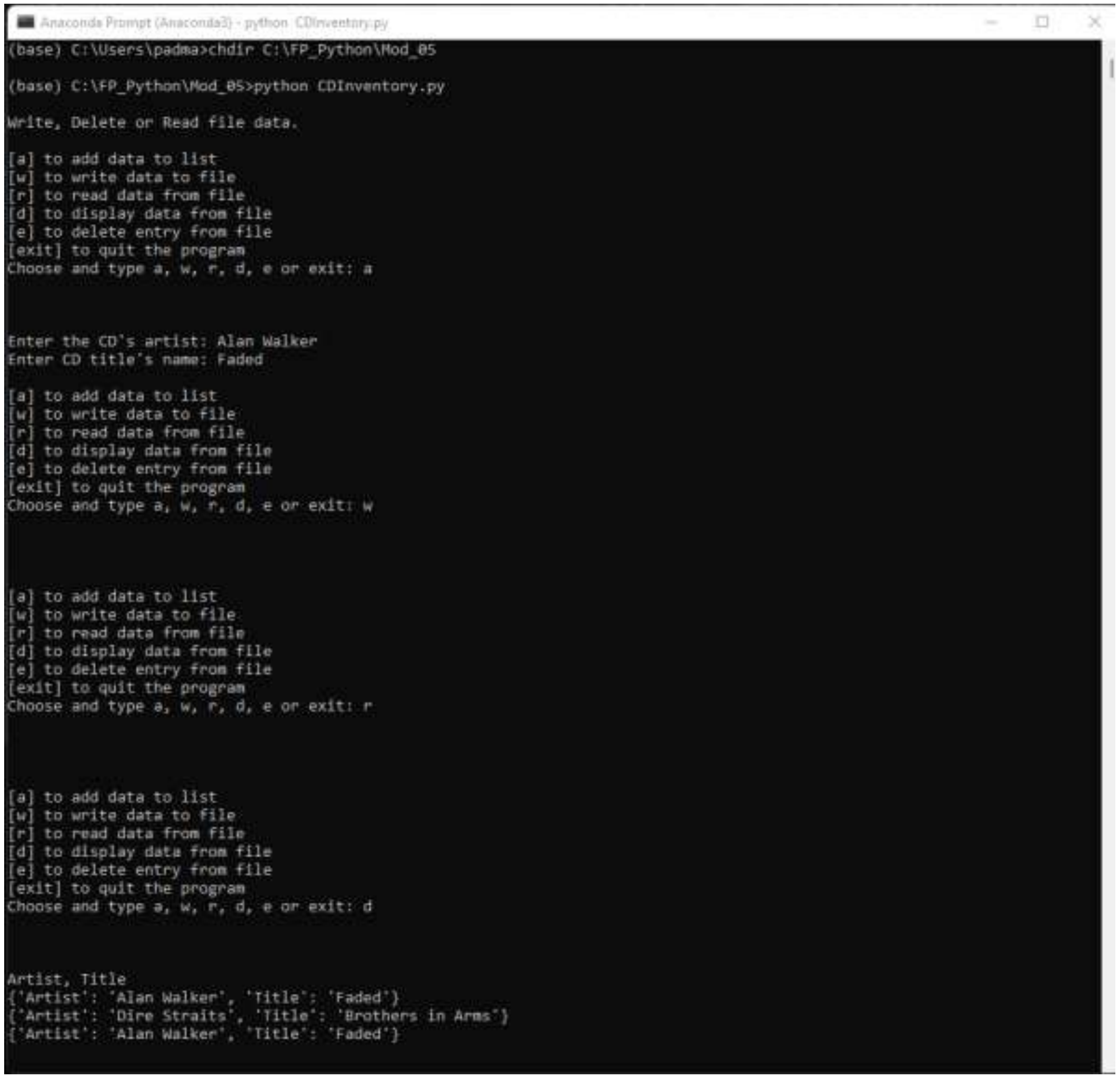


```
In [7]: runfile('C:/FP_Python/Mod_05/CDInventory.py', wdir='C:/FP_Python/Mod_05')

Write, Delete or Read file data.

[a] to add data to list
[w] to write data to file
[r] to read data from file
[d] to display data from file
[e] to delete entry from file
[exit] to quit the program

Choose and type a, w, r, d, e or exit: 1



Please choose one of a, w, r, d, e or exit!

[a] to add data to list
[w] to write data to file
[r] to read data from file
[d] to display data from file
[e] to delete entry from file
[exit] to quit the program

Choose and type a, w, r, d, e or exit: exit
```

Figure 6: Exit option and Invalid input given

Execution on Terminal

Here too, while the following image captured shows a successful execution on the Terminal, since the program was not closed and / or the list of data was read first before displaying it, the newly-added entry is shown as a duplicate.



```
(base) C:\Users\padma>chdir C:\FP_Python\Mod_05

(base) C:\FP_Python\Mod_05>python CDInventory.py

Write, Delete or Read file data.

[a] to add data to list
[w] to write data to file
[r] to read data from file
[d] to display data from file
[e] to delete entry from file
[exit] to quit the program
Choose and type a, w, r, d, e or exit: a



Enter the CD's artist: Alan Walker
Enter CD title's name: Faded

[a] to add data to list
[w] to write data to file
[r] to read data from file
[d] to display data from file
[e] to delete entry from file
[exit] to quit the program
Choose and type a, w, r, d, e or exit: w




[a] to add data to list
[w] to write data to file
[r] to read data from file
[d] to display data from file
[e] to delete entry from file
[exit] to quit the program
Choose and type a, w, r, d, e or exit: r




[a] to add data to list
[w] to write data to file
[r] to read data from file
[d] to display data from file
[e] to delete entry from file
[exit] to quit the program
Choose and type a, w, r, d, e or exit: d



Artist, Title
{'Artist': 'Alan Walker', 'Title': 'Faded'}
{'Artist': 'Dire Straits', 'Title': 'Brothers in Arms'}
{'Artist': 'Alan Walker', 'Title': 'Faded'}
```

Figure 7: Execution of code on Terminal

GitHub

The following is the link to my GitHub repository for this assignment:

https://github.com/PadmaBham/Assignment_05.git

Appendix

1. My **Python script** for the assignment is as follows:

```
#------------------------------------------#
# Title: CDInventory.py
# Desc: Inventory of CDs that user can add to, delete from or display
# Change Log: (Who, When, What)
# DBiesinger, 2030-Jan-01, Created File
# PBhamidipati, 2022-Nov-11, Edited File to add functionality to add, write, read and display menu items, including code to read and write date from list
to file and vice versa
# PBhamidipati, 2022-Nov-12, Edited File to use dictionaries, making the data structure from a list of lists to list of dictionaries
# PBhamidipati, 2022-Nov-13, Edited File to add delete option to the menu and its functionality; impacts ln#19, #22
#------------------------------------------#

# Declare variables

strChoice = '' # User input
lstTbl = []
strFileName = 'CDInventory.txt'  # data storage file
objFile = None  # file object

# Get user Input
print('\nWrite, Delete or Read file data.')
while True:
    print('\n[a] to add data to list\n[w] to write data to file\n[r] to read data from file')
    print('[d] to display data from file\n[e] to delete entry from file\n[exit] to quit the program')
    strChoice = input('Choose and type a, w, r, d, e or exit: ').lower()  # convert choice to lower case at time of input
```

```
print('\n\n')

strChoice = strChoice.strip()

if strChoice == 'exit':
    break

if strChoice == 'a':  # no elif necessary, as this code is only reached if strChoice is not 'exit'
    dictRow = {} # initialise variable to hold dictionary data

    # Ask user to input data and store it as a dictionary
    dictRow['Artist'] = str(input('Enter the CD\'s artist: ')) # ask Artist Name
    dictRow['Title'] = str(input('Enter CD title\'s name: ')) # ask CD Title

    # Add data to list in memory
    lstTbl.append(dictRow)

elif strChoice == 'w':

    # List to File
    # Write from in-memory list to file. (convert dictionary to string)
    for row in lstTbl:
        strRow = ''
        for key in row:
            strRow += row[key] + ','
        strRow = strRow[:-1] + '\n'
        objFile = open(strFileName, 'a')
        objFile.write(strRow)
        objFile.close()

elif strChoice == 'r':

    # File to read
    objFile = open(strFileName, 'r')

    # Read the file line-by-line into in-memory list. (create back the dictionary type)
    for row in objFile:
        rdStrRow = row.strip().split(',')
        createDictRow = {'Artist': str(rdStrRow[0]), 'Title': str(rdStrRow[1])}
        lstTbl.append(createDictRow)
        objFile.close()

    elif strChoice == 'd':

        # Display data
        print('Artist, Title')

        # Display the data to the user.
        for row in lstTbl:
            print(row) # observation: if data is displayed before closing and restarting the program, then data maybe duplicated based on order of reading and
displaying

    elif strChoice == 'e':

        # Get the name of Artist from user, whose entry is to be deleted
        delArtist = str(input('Enter Artist\'s name for entry deletion: '))

        # File to read
        objFile = open(strFileName, 'r')
        detInd = 0 # initialise a counter to determine the index of the entry to be deleted

        # Read the file line-by-line into in-memory list. (create back the dictionary type)
        for row in objFile:
            rdStrRow = row.strip().split(',')
            createDictRow = {'Artist': str(rdStrRow[0]), 'Title': str(rdStrRow[1])}
            lstTbl.append(createDictRow)
        objFile.close()

        for item in lstTbl:
            if delArtist == item['Artist']:
                break
            else:
                detInd +=1

        if detInd < len(lstTbl):
            del lstTbl[detInd]
            objFile = open(strFileName, 'w') # so the file is overwritten
            objFile.truncate()

            for row in lstTbl:
                strRow = ''
                for key in row:
                    strRow += row[key] + ','
                strRow = strRow[:-1] + '\n'
                objFile.write(strRow)
            objFile.close()

        else:
            print('Please choose one of a, w, r, d, e or exit!')
```

2. Reference material to learning this module has been all the reading documentation and videos demonstrations that were provided in this course.

3. Search strings for this week and URLs referenced:
   - ➢ how to delete an item from a list type in python
     - URL: www.geeksforgeeks.com/
   - ➢ file overwrite mode from in python
     - URL: www.geeksforgeeks.com/